DATA WAREHOUSING THROUGH MULTI-AGENT SYSTEMS IN THE MEDICAL ARENA

António Abelha, José Machado, Victor Alves and José Neves
Departamento de Informática
Universidade do Minho
Braga, Portugal
{abelha,jmac,valves, jneves}@di.uminho.pt

Abstract – In this paper it is presented AIDA, an Agency for Integration, Archive and Diffusion of Medical Information. It configures a data warehouse, developed using Multi-Agent technology, that integrates and archives information from heterogeneous sources of a health care unit. AIDA is like a symbiont, with a close association with core applications at any health care facility, namely the Picture Archive Communication System, the Radiological Information System and the Electronic Medical Record Information System, that are built upon pro-active agents and communicate with the AIDA's ones.

Keywords: Multi-Agent Systems, Data Warehousing, Medical applications.

I. INTRODUCTION

Medical Information Systems (MIS) are seen as a way of optimising the use of the existing health care infrastructure, without resorting to new and costly building construction. The qualitative design of such an environment requires a basic understanding of patient and doctors related characteristics and capabilities. Patient care, patient education, medical education, and clinical research need to be considered to meet the basic requirements on the level of desirable services, determined on the basis of the patient's length of stay; i.e., used for modelling the significant entities of such a world. The aim is to extract conclusions for the level of services provided to the users. One's concept will capture, as well as will integrate, the basic design principles under which MIS may be set.

Medicine has been for some years a very attractive domain for Computer Science (CS) researchers, in general. There is a great potential for information automation, and a lot remains to be done. Medical Informatics (MI) is indeed becoming an issue of study in which Medicine and Computing overlap. Another reason for this increasing interest is costs. Today's strained healthcare economics makes it necessary for expensive resources to be efficiently used, and requires a balanced management. CS researchers have long used Medicine to elaborate on their own work. The field is probably one of the most knowledge intensive ones, loaded with human reasoning, with most of the procedure relying exclusively on the clinical experts. This makes healthcare a perfect target for CS, since conventional systems are naturally bounded by their lack of rich knowledge representation and proof schemes. Medicine allows for the testing and exposing of new ideas and techniques by CS scientists, which creates some complicity between the two communities.

What is the best approach to help doctors and patients? It is believed Artificial Intelligence (AI) enters at this point. A system intended to be a support tool for clinicians must have its focus on knowledge representation and reasoning schemes. Such system should be able to explain and justify its conclusions. On the other hand it has to be flexible enough to allow for simulation, training of novices and maintenance [1].

II. AIDA

Information sources in a healthcare unit are distributed, heterogeneous, large and complex; integrate medical equipments that speak different languages; are built around information systems customized by several companies using different operating systems, languages, applications and hardware. Communications are sometimes limited by old infrastructures and new projects collide with financial restrictions and bureaucratic delays. The homogeneity of clinical, medical and administrative systems is not possible due to financial and technical restrictions, as well as functional needs. The solution is to integrate, diffuse and archive this information under a dynamic framework, in order to share this knowledge with every information system that needs it. AIDA – Agency for Integration, Diffusion and Archive of Medical Information [2] - is an agency that provides intelligent electronic workers, here called pro-active agents, and in charge of tasks such as communicating with the heterogeneous systems, sending and receiving information (e.g., medical or clinical reports, images, collections of data, prescriptions), managing and saving the information and answering to information requests, with the necessary resources to their correct and in time accomplishment (Figure 1). AIDA also supports Web based services to facilitate the direct access to the information and communication facilities set by the humans; i.e., AIDA's construction follows the acceptance of simplicity, the conference of the achievement of common goals and the addressing of responsibilities; the main goals are to integrate, diffuse and archive large sets of information from heterogeneous sources (de-

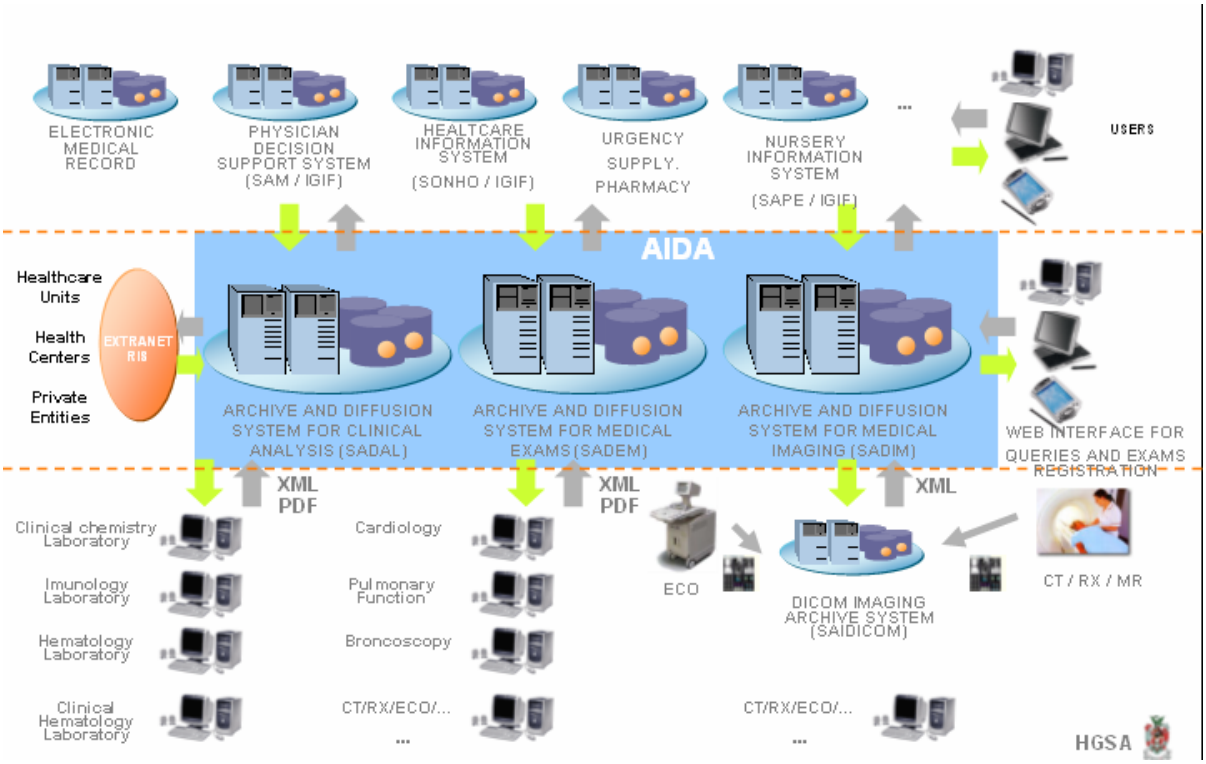

Figure 1- The AIDA modules

partments, services, units, computers, medical equipments); AIDA also provides tools in order to implement communication with human agents based on web based services. Under these presuppositions, a Healthcare Information System (HIS) will be addressed in terms of:

- The Administrative Information System (AIS), which intends to represent, manage and archive the administrative information during the episode (an episode is a collection of all the operations assigned to the patient since the beginning of the treatment until the end);
- The Medical Support Information System (MIS), which intends to represent, manage and archive the clinical information during the episode;
- The Nursing Support Information System (NIS), which intends to represent, manage and archive the nursing information during the episode;
- The Electronic Medical Record Information System (EMR);
- The Information Systems (DIS) of all the departments or services, in particular of the laboratories (Labs), Radiological Information System (RIS) and Medical Imaging (PACS - Picture Archive and Communication System), which deals with images in a DICOM format [3].

In order to implement this system, distributed by nature, intranet technology was used on the side of the end user. All the system interfaces are user-friendly, Web based and low cost. It uses freeware tools or software database packages which licenses belong to the Portuguese Health Ministry (e.g., Oracle software). The intranet was implemented using computers with *LINUX* as operating system. Communication among agents will be specified by defining the transportation (i.e., the process that allows for a message to be sent or received), the language (i.e., the meanings of a message), the ontology (i.e., the conversational structure) and the architecture (i.e., the system links according to given protocols).

A message is sent by an agent using XML [4]. According to the ontology, the message is processed, integrated and archived in large databases. The ontology is defined by the administrators and can be managed using web tools. The healthcare network is used for physical transportation.

XML is a meta-markup language and stands for Extensible Markup Language. It is a set of rules to define semantics tags in documents, which are broken into parts. The language defines a syntax in which other field-specific markup languages can be written (e.g., HTML and TeX). HTML and TeX define a fixed set of tags that describe a fixed number of elements. In XML, data is not formatted to fit into paragraphs, text fields, list items, table cells or others general categories. Tags can be documented in a Document Type Definition (DTD); i..e, a vocabulary and a syntax for some kinds of documents. XML has an excellent format for the interchange of data among different applications be-

cause it is free to use and easy for both human beings

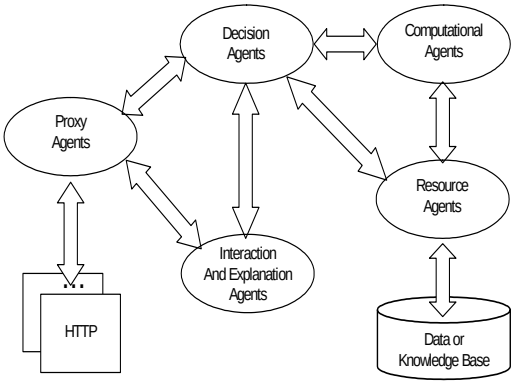

Figure 2- The PACS agents

and computers to read and write. It is an obvious choice for exchange and share data, programs or languages between agents. XML also provides a client-side include mechanism that integrates data from multiple sources and displays it as a single document, what is very useful when working with large information repositories like RIS, PACS, EMR and AIDA. RDMBS applications (e.g, Oracle, MySQL, Microsoft) provide multiple programs to create, extract and parse XML documents from databases using Perl, Java, PHP or any other convenient language. AIDA uses XML documents to communicate between external data sources and operational databases. XML is also embedded in PHP programs that provide user interfaces.

The intelligence of the system as a whole arises from the interactions among all the system's components. The interfaces are based on Web-related front-ends that can be accessed using a standard Web browser, querying or managing the data warehouse [5][6]. Such an approach can provide decision support (e.g., with the radiologist conducting a form of dialogue with the technicians to extract knowledge and test hypothesis, using a strategy to compare a modality independent model with the image via an intermediate symbolic feature space).

III. MULTI-AGENT SYSTEMS

Multi-agent Systems (MAS) set a new paradigm in problem-solving via theorem proving; i.e., agent-based computing has been hailed as a significant breakthrough in problem solving and/or a new revolution in software development and analysis. Indeed, agents are the focus of intense interest on many sub-fields of Computer Science, being used in a wide variety of applications, ranging from small systems to large, open, complex and critical ones; i.e., agents are not only a very promising technology, but are emerging as a new way of thinking, a conceptual paradigm for analysing problems and for designing systems, for dealing with complexity, distribution and interactivity, may be a new form of computing and intelligence.

Although there is no universally accepted definition of agent, in this work such an entity is to be understood as a computing artefact, being it in hardware or software, that exhibit the properties:

- *autonomy*; i.e., whereby such entities have the ability to act without the direct intervention of their peers, namely humans;
- *reactivity*; i.e., whereby such entities are situated in an environment that can perceive through sensors and act in reaction to stimuli (e.g., revising their beliefs according to or in reaction to new inputs);
- *pro-activity*; i.e., whereby such entities exhibit intelligent problem solving capabilities (e.g., planning their activities in order to achieve short or long term goals); and
- *social behaviour*; i.e., whereby such entities are *aware* of one another, can *interact* with one another and may *modify* their behaviour in response to others; can *communicate* via a set of low or high level constructs and protocols as well as means of addressing and direct communication; can *cooperate* in order to achieve joint as well as individual goals, what means that must have the ability to negotiate with other agents either to accomplish their own goals or to joint plans to achieve common goals; to perform *belief revision* in the context of additional sources of information provided by their peers.

To develop such systems, a standard specification method is required, and it is believed that one of the keywords for its wide acceptance is simplicity. Indeed, the use of intelligent agents to simulate human decision making in the medical arena offers the potential to set an appropriate software development and analysis practice and design methodology that do not distinguish between agent and human, until implementation. Being pushed in this way, the design process, the construction of such systems, in which humans and agents can be interchanged, is simplified; i.e., the modification and development in a constructive way, of multi-agent healthcare systems with a human-in-the-loop potential aptitude is becoming central in the process of agent-oriented software development and analysis. These systems have provided a clear means of monitoring the agent's behaviour with significant impact in their process of knowledge acquisition and validation. MAS are a natural connection to intelligent systems evolution, being elements for task substitution or delegation, usually performed by human beings. However agent based systems have some restrictions, such as global system control and universal view absences, and some want of confidence and fear of competence delegation by human beings [7]. To delegate tasks, bilateral confidence relations have to be established. Organizations may also mature their experience relatively to the use of autonomous software components.

Agent theory in Artificial Intelligence deals with the behaviour of autonomous and rational intelligent enti-

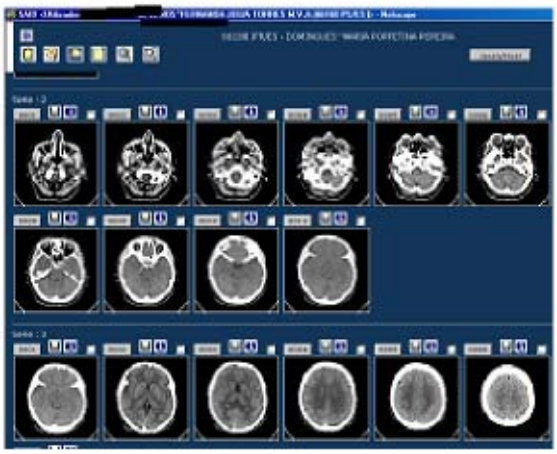

Figure 3- The PACS web interface

ties which are capable of executing actions in order to reach a given goal. Their behaviour depends on intentional concepts like knowing, wanting, liking or believing, on actions that are executed depending on the states of the system, the level of the learning procedures and the context under which the conditions are tested. Indeed, actions are related to knowledge, the state of the world is perceived and plans for actions are formed. Actions may be periodic, spontaneous and may benefit the user. Agents may also take their own initiatives ad may also be integrated on the World Wide Web (WWW) (Figure 3)(Figure 4). The problem is that the WWW is client-server oriented and WWW agents use peer-to-peer communication, so multiple messages from agents and collaborative tasks are not directly enabled. The WWW is oriented around structures representing transport and display of information, in particular in Hyper Text Markup Language (HTML). With Extended Logic Programming coupled with Java based implementations of the Logic Programming paradigm, it is possible to conceive agents that send messages and interact with people or other agents through browsers, aiming at distributed interactive simulation environments; i.e., any complex software system can be viewed as a collection of independent cooperating units or entities, each of which implements a subset of an application's total functionality [8][9][10][11].

IV. PICTURE ARCHIVE COMMUNICATION SYSTEMS

The computational architecture that supports *Medical Information* in the area of *Imagiology* is an example of an external data source for AIDA. To get the knowledge required to construct these system, one must be involved and familiarized with the DICOM standard, Computer Tomography (CT), Magnetic Resonance (MRI), procedure documentation, debriefing of Neurologists. It also involves the regular participation of the authors in meetings at the premises of the healthcare unit [12][13]. Architecture has then been envisaged to support the medical applications in terms of *Intelligent Medical Information Systems*, a form of a web spider of an intelligent information processing system, its major subsystems, their functional roles, and the flow of information and control among them, with adjustable autonomy. Indeed, many complex systems are made up of specialized subsystems which are understood as intelligent entities or agents that interact in flexible, goal-directed manners, and are understood as theories; i.e., the intelligence of such a system as a whole arises from the interactions among all the system's entities. There is also a purely communicative *Multi-Agent System* (*MAS*); i.e., there is no external environment influence and the agents communicate only through messages. The general architecture of such systems is given in Figure 2. A brief description of the different types of agents involved is given below:

- The Proxy Agents (PAs) which provide the bridges between the users and the system in terms of questions that may be formulated, explanations that may be required, decisions that may have to be taken and/or view of final results. The system's interfaces are based on Web-related front-ends using HTML pages, that can be accessed using a standard Web browser;
- The *Decision Agents* (*DAs*) which present mediating capabilities, act by accepting a task from the *PAs*. They may decompose it in sub-tasks, sending them to be processed in the *CAs*, and integrating the results (returned by the *CAs*);
- The *Computing Agents* (*CAs*) which accept requests for specific tasks from the *DAs*, returning the results;
- The Resource Agents (*RAs*) which present all the necessary information to their peers in order to allow them to access a specific data resource; and
- The *Interaction and Explanation Agents* (*IEAs*) which act based on argumentative proceedings which are fed with data and/or knowledge coming either from the *PAs* or the *Das* (note that the execution plans received from the *DAs* may be partial, so that only upon a completion of a task a trace can be compiled and an explanation can be delivered to the *PAs* and/or *DAs*).

Since their introduction, *PACS* have attracted a mixed reaction, mingling excitement at their potential with dissatisfaction with their various limitations. This has been true both in general *AI*, as in *The Medicine* field. Much work has therefore been concentrated on understanding these drawbacks, and developing problem solving methods and building strategies for such systems that may mitigate such faults. The overall trend has been away from seeing the process as one of encoding heuristics derived from an expert, towards modeling the domain on which the expertise operates. Further, it has become recognized that modeling requires as a

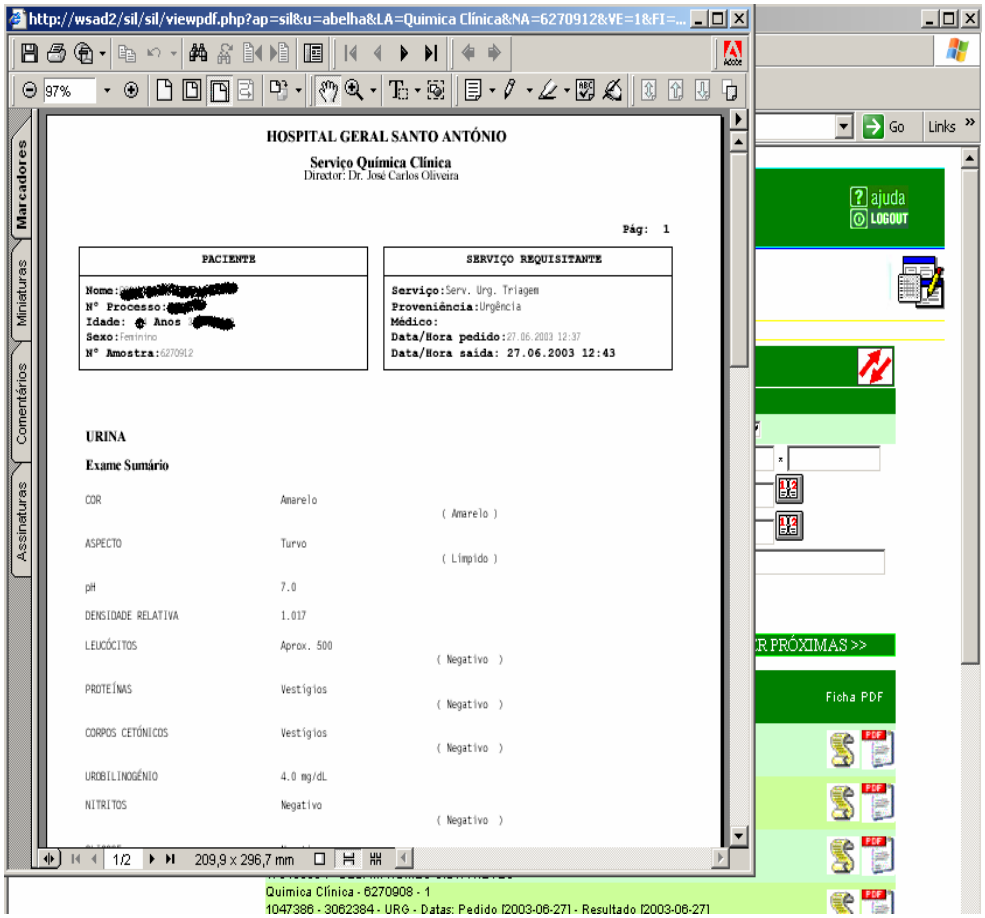

Figure 4- A Web Interface

precondition that the domain be conceptualized, and that the assumptions incorporated in the conceptualization must be made explicit.

However, if agent technologies are to be effective, design issues need careful consideration; i.e., how are *MAS* architectures to be designed for maxim effect?; can such architectures be used to support and enhance existing work practices?; what kind of information processing work and for what medicine's domains?; how can such systems be design to successfully complement people's existing practices and preferences? Indeed, what can be done with this conceptualization is now well understood in terms of its limitations, in the sense of the sort of medical domains to which it can be applied; its restrictions upon the duration or extent of an estate in the sense of the sort of system that will result from its application; and how to build and use *PACS.*

The system is characterised by the use of explicit anatomical models and for the visualisation of the anatomical structures identified in the image segmentation. The anatomical model makes a major component of the system, and is organised in terms of a semantic network. The inference engine handles the decision making proceedings during the process of segmenting major anatomical landmarks.

The DICOM image server supports the medical interface – this window sets the via for the visualisation and exploration of original DICOM data from CT, MRI, and so on. It provides the user with interactive image visualisation functions, like gray level windowing.

V. DATA WAREHOUSE FOR HEALTHCARE

Relational Database Management Systems (RDBMS) are widely used by healthcare units for maintaining data that documents everyday operations [14]. The applications that update such operational data or transactions make typically small changes and a large number of transactions. RDBMS have been optimised to perform reliably and efficiently those operations, usually named online transaction processing (OLTP) applications (i.e., applications that support many users executing small transactions) [15].

Current and historical data has also to be comprehensively analysed and explored, identifying useful trends and creating summaries, to support decision making.

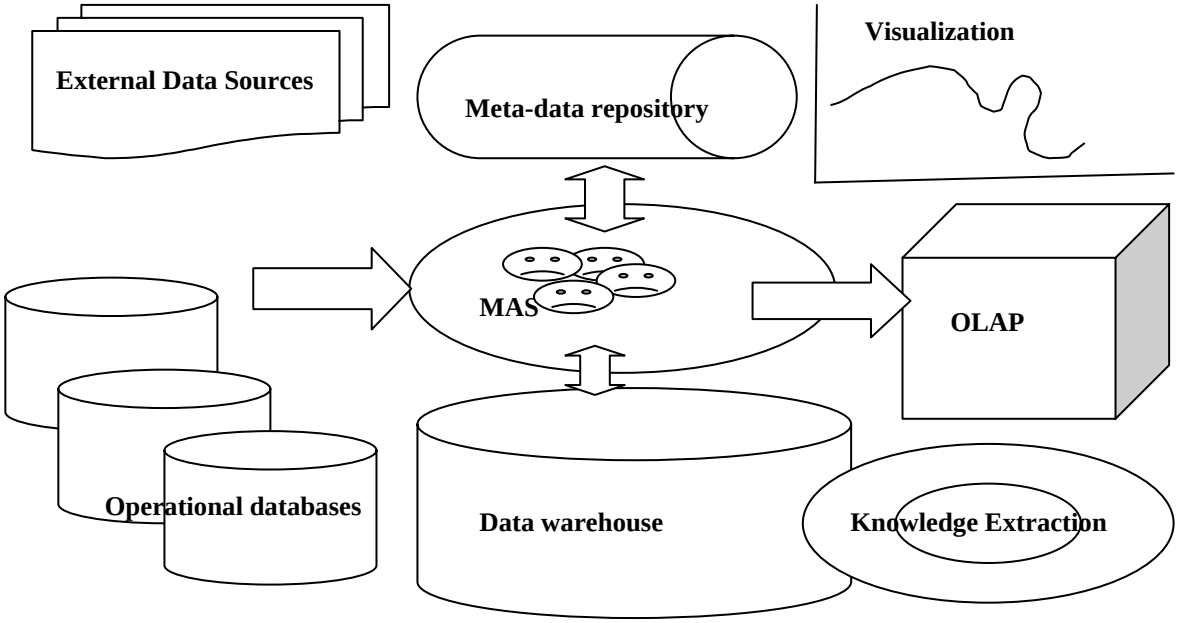

Figure 5 - A data warehouse typical architecture using MAS

These operations are referred as decision support applications. On the other hand, and in the last few years, SQL has been extended with new constructs to support it, performing new indexing and query optimisation techniques to run fast complex queries. For example, one technique that has gained popularity in applications involving complex data access is the use of views. Queries on views can be answered by evaluating the view definition when the query is submitted and precomputing the view definition can decrease answer time. Healthcare units can consolidate information from several databases into a data warehouse by receiving data from many heterogeneous sources into a location or materializing a view defined over several tables. Those views are created, managed and used by specialized agents of the AIDA environment (Figure 5).

SQL queries are frequently embedded in a host language program. Many queries involve conditions over time or require aggregating over time periods. SQL provides poor support for such time series analysis but with the help of autonomous agents this support may improve.

Users often need to make several related queries. Users have to write a collection of independent queries since there is no possible way to express the commonly occurring families of queries. RDBMS has no way to recognize and exploit optimisation opportunities arising from executing many related queries together. Here, agent technology may give special support in order to help users. Three broad classes of analysis tools are available:

- Some systems support a class of stylised queries that involve group-by and aggregation operators and provide support fro complex boolean conditions, statistical functions and features for time series analysis. These applications are called Online Analytic Processing (OLAP) [16]. In these systems, the data is thought as a multidimensional array and is influenced by end-user tools, such as spreadsheets, in addition to database query languages.
- Some RDBMS that support traditional SQL-style queries, are also designed to support OLAP queries efficiently. Such systems stand for RDBMS optimised for decision support applications.
- The use of analysis tools is motivated by the desire to find interesting or unexpected trends and patterns in large data sets, rather than the query portfolio just listed. The goal of knowledge extraction is to support exploratory analysis over very large data sets. Evaluating OLAP or knowledge extraction queries over globally distributed data is likely to be slow. The natural solution is to create a centralized repository of all the data; i.e., a data warehouse managed by a MAS.

On the other hand, a large database is usually composed by small code tables that rarely increase in size, by large business transaction tables that account for the majoring of the records and increase in size over time, by aggregation tables that depend on the application design and are based on aggregations of data from the large transaction ones, and temporary tables to load data for processing and bulk data manipulation.

Meta-data pointing to external files can be created based on the meta-data definitions. Data is extracted from operational databases and external sources, cleaned to minimize errors and fill in missing information when possible or available, and transformed to reconcile mismatches. Transforming data is typically

accomplished by defining a relational view over the tables in the data sources, materializing such views and storing them in the data warehouse, that is different from the database(s) containing the tables it is defined over. The cleaned and transformed data is finally loaded into the data warehouse. Additional pre-processing such as sorting and generation of summary information is carried out at this stage. Data is partitioned and indexes are built for efficiency. Loading is a slow process and parallelism is important for reduce integration time. After loading data, the data is periodically refreshed to reflect updates and old data is periodically purged (e.g., to archive media). Maintaining replicas of distributed source relations is an essential part of warehousing, even if violating the principle of distributed data independence. The system catalogs are very large and often stored and managed in a separate database called metadata repository (e.g., in addition to describe the fields of a table it must be keep track of the source of each warehouse table). The data of a warehouse is accessed and analysed using a variety of tools, including OLAP query engines, knowledge extraction algorithms, information visualization tools, statistical packages and report generators.

A data warehouse occupying a terabyte of disk is now quite common. But it can become unresponsive, unstable and unusable. Oracle general purpose database management product has not always had full market recognition for its advance in scalability [12]. Well informed professionals have come to believe that Oracle is not suited to data warehouse requirements involving more than a terabyte of disk. In fact, Oracle is in successful operation with rapidly growing data warehouses much larger than a terabyte. Oracle is capable of managing data warehouses containing multiple terabytes of data and handling thousands of users. In fact, the evidence shows that Oracle has a significant presence in the management of very large data warehouses, and there are customers that are moving forward to use Oracle on a yet larger scale. In a distributed environment, database on separate servers may be accessed during a single transaction or query. Oracle can be configured to support remote queries or replication strategies where the databases can serve as the sources for data diffusion. Materialized views are objects to summarize, pre-compute, synchronize or distribute data. Aggregate information is stored to increase the performance of the answering systems.

The size of the database is not the only important factor in such assessments: the implementer must also consider the nature of the use, the application, the workload, the performance requirements and other factors (e.g., security).

Some problems may be identified. Data size matters (50 GB is not the same than 1 TB) and data volume is an important parameter to take under consideration when dimensioning the data warehouse and developing agents. The consistent availability and good performance in the presence of continuing demanding and changing use, interfere with the system stability. Indeed, in large organizations with thousands of potential users, data warehouses may be available to business partners on extranets and to customers on internet. A growing set of applications sharing the data must be available and data warehouse may be adaptative; i.e., to deal with rapid change in business environments, strategies and activities, data warehouses must change rapidly as well, new applications are implemented, changing workload and usage patterns, and queries are focused on a different subject; acquisitions occur. At the same time, volume grows, more tables are created, more queries are submitted by more users that generate more demand. The data warehouse become larger and more complex, the difficulty of managing it rises, and these requirements can be addressed only partly by the creation of external tools and management facilities.

A data warehouse must be addressed by building manageability into the database engine and by making facilities, and reducing the staff of database administrators to maintain it. It must support a large population of users where the data volume is the challenge. Data warehouse is there to help manage the business system and must also deal with changing worlds or environments. It must support at the same time small transactions, executed by users, and very large transactions, executed by software agents, during the loading of data into the data warehouse, processes that can occur at times when online warehouse users are accessing the database, searching online updated records.

VI. RADIOLOGICAL INFORMATION SYSTEM

Radiological Information System (RIS) is a core application that is built around a MAS that receives information from heterogeneous sources (e.g., administrative applications, agendas, medical equipments, AIDA, PACS) and integrate it into the data warehouse in order to be managed by other external applications. Patient scheduling is made in real time and the integration with radiological equipments is made using DICOM worklist protocol through the MAS.

VII. ELECTRONIC MEDICAL RECORD

Electronic Medical Record (EMR) is also a core application which covers horizontally the health care unit and make possible a transverse analysis of medical records along the several services, units or treated pathologies. Beyond the organizational, functional, technical and scientific requisites, ones have to attend ethical and legal ones, as well as data quality, information security, access control and privacy. An Electronic Medical Record is an assembly of standardized documents, ordered and concise, directed to the register of actions and medical procedures; a set of information compiled by physicians and others health professionals;

a register of integral facts, containing all the information regarding patient health data; and a follow up of the risk values and clinical profile.

VIII. CONLUSIONS AND FUTURE WORK

The system is now at work at the Hospital Geral de Santo António, in Oporto, one of the two major healthcare facilities in the north of Portugal. It is expected that under the *Interreg III* programme, an *European Commission* initiative, it will spread across the health care facilities in the north of *Portugal*, as well as in *Galiza* and *Castela and Leão* in *Spain*.

REFERENCES

[1] Abelha, A., Multi-Agent Systems to support Cooperative Work in Helath Care Units, PhD Thesis, Departamento de Informática, Universidade do Minho, Braga, Portugal, 2004.

[2] Abelha A., Machado M., Santos M., Sollari A., Rua F., Paiva M. and Neves J., Agency for Archive, Integration and Diffusion of Medical Information, Proceeding of AIA'2003, IASTED International Conference on Artificial Intelligence, Benálmadena, Spain, 2003.

[3] Alves V., Neves J., Maia M., Nelas L., A Computational Environment for Medical Diagnosis Support Systems. ISMDA2001, Madrid, Spain, 2001.

[4] Harold E., XML Bible – Gold Edition, Hungry Minds Inc., NY, 2001.

[5] Chaudhuri S. and Dayal U., An Overview of Data Warehousing and OLAP Technology, SIGMOD record, 26(1), 1997.

[6] Kimball R., The Data Warehouse Toolkit, John Wiley and Sons, 1996.

[7] Heinze, C., Papasimeon, M., and Goss, S., Specifying Agent Behavior with Use Cases, in Design and Applications of Intelligent Agents – Proceedings of the Third Pacific Rim International Workshop on Multi-agent Systems PRIMA 2000, eds. C. Zhang and V. Soo, 128-142 (Lecture Notes in Artificial Intelligence, 1881).

[8] Neves, J., Machado, J., Analide, C., Novais, P., and Abelha, A. "Extended Logic Programming applied to the specification of Multi-agent Systems and their Computing Environment", in Proceedings of the ICIPS'97 (Special Session on Intelligent Network Agent Systems), Beijing, China, 1997.

[9] Neves, J. "A Logic Interpreter to handle time and negation in logic data bases", in Proceedings of the ACM 1984 Annual Conference, pages 50-54, San Francisco, California, USA, 1984.

[10] Alves, V., Neves, J., Maia, M. and Nelas. L., Computer Tomography based Diagnosis using Extended Logic Programming and Artificial Neural Networks. Proceedings of the International NAISO Congress on Information Science Innovations ISI2001, Dubai, U.A.E., 2001.

[11] Traylor, B, and Gelfond, M. "Representing null values in logic programming", in Proceedings of the 1993's International Logic Programming Conference Symposium (ILPS'93), pages 35-47, Vancouver, British Columbia, Canada, 1993.

[12] Neves, J., Alves V., Nelas L., Maia M., and Cruz R. "A Multi-Feature Image Classification System that Reduces the Cost-of-Quality Expenditure", in Proceedings of the Second ICSC Symposium on Engineering of Intelligent Systems, Paisley, Scotland, UK, pages 594-554. 2000

[13] Looney K. and Theriault M., DBA Handbook, Oracle Press, Mc Graw Hill / Osbourne, 2002.

[14] Ramakrishman R. and Gehrke J., Database Management System, 3rd Edition, Mc Graw Hill Higher Education, 2003.

[15] Codd E., Twelve Rules for On-line Analytic Processing, Computerworld, 1995.

[16] Shoshami A., OLAP and Statistical Databases: Similarities and Differences, in ACM Symposium on Principles of Database Systems, 1997.